(19) BUNDESREPUBLIK DEUTSCHLAND

DEUTSCHES PATENTAMT

(12) Übersetzung der europäischen Patentschrift

(87) EP 0 384 610 B1

(10) DE 690 28 226 T 2

(51) Int. Cl.[6]: G 06 F 1/00
G 06 F 12/14

DE 690 28 226 T 2

(21) Deutsches Aktenzeichen: 690 28 226.5
(86) Europäisches Aktenzeichen: 90 301 253.2
(86) Europäischer Anmeldetag: 6. 2. 90
(87) Erstveröffentlichung durch das EPA: 29. 8. 90
(97) Veröffentlichungstag der Patenterteilung beim EPA: 28. 8. 96
(47) Veröffentlichungstag im Patentblatt: 13. 3. 97

(30) Unionspriorität: (32) (33) (31)
24.02.89 US 315071

(73) Patentinhaber:
International Business Machines Corp., Armonk, N.Y., US

(74) Vertreter:
Teufel, F., Dipl.-Phys., Pat.-Anw., 70569 Stuttgart

(84) Benannte Vertragstaaten:
DE, ES, FR, GB

(72) Erfinder:
Enescu, Michael Alexander, Sunnyvale, CA 94087, US; Lum, James, Redwood City, CA 94065, US

(54) Gegen unbefugte Manipulation gesichertes Zugangsberechtigungsverfahren

DE 690 28 226 T 2

B E S C H R E I B U N G

GEGEN UNBEFUGTE MANIPULATION GESICHERTES ZUGANGSBERECHTIGUNGSVERFAHREN

Die vorliegende Erfindung betrifft die Sicherheit von Computersystemen und insbesondere ein gegen unbefugte Manipulation gesichertes Zugangsberechtigungsverfahren zur Kontrolle des Zugangs von Programmen, Prozessen oder Benutzern zu Ressourcen, die von einem Computersystem definiert werden.

Hierbei ist Bezug zu nehmen auf Peterson und Silberschatz, "Operating System Concepts", Copyright Addison-Wesley Publishing Co. 1983, Kapitel 11, S. 387-419, zum Thema Schutz, und auf Dorothy Denning, "Cryptography and Data Security", Copyright Addison-Wesley Publishing Co. 1982, Kapitel 4, S. 209-230, zum Thema Zugangskontrollen.

Diese Referenzen beschreiben Mechanismen zur Kontrolle des Zugangs von Programmen, Prozessen oder Benutzern zu Ressourcen, die von einem Computersystem definiert werden. Sowohl Peterson als auch Denning favorisieren offenbar eine entweder statisch oder dynamisch implementierte Zugangsmatrix als Schutzkonstrukt der Wahl in solchen Systemen.

Das Matrixkonstrukt verwendet Reihen zur Darstellung von Domänen und Spalten zur Darstellung von Objekten. Jeder Eintrag in der Matrix besteht aus einer Menge von Zugangsrechten. Wenn ein Computer eine globale Tabelle enthielte, die aus einer Menge geordneter Tripel

<Benutzer(i), Objekt (j), Rechtemenge (k)> bestünde, würde bei jeder Ausführung einer Operation M an einem Objekt O(j) durch einen Benutzer U(i) eine Suche nach dem Tripel <U(i),O(j),R(k)> durchgeführt, und die Operation könnte nur fortgesetzt werden, wenn beim Vergleich eine Entsprechung gefunden wird.

Beide Referenzen beschreiben ferner mehrere Konstrukte, die von einer Zugangsmatrix abgeleitet sind. Hierzu zählen Zugangslisten, Fähigkeitslisten sowie Schloß-Schlüssel-Mechanismen. Es muß berücksichtigt werden, daß eine Zugangsliste listenorientiert ist, eine Fähigkeitsliste ticketorientiert ist und ein Schloß-Schlüssel-Mechanismus Merkmale von beiden vereint.

Eine Zugangsliste ist nicht mehr als eine Menge geordneter Paare <U(i), R(k)>, die nach den einzelnen Objekten O(j) sortiert sind. Eine Fähigkeit ist ein Ticket, das jedem Inhaber die Zugangsrechte R für das Objekt O verleiht. Der bloße Besitz bedeutet, daß der Zugang erlaubt ist.

Bei einem Schloß-Schlüssel-Mechanismus enthält jedes Objekt O(j) ein einmaliges Bitmuster, das als "Schloß" bezeichnet wird, während nur bestimmte Benutzer im Besitz eines einmaligen Bitmusters sind, das als "Schlüssel" bezeichnet wird. Somit kann ein U(i) nur dann einen Schlüssel zu O(j) bekommen, wenn er Zugangsrechte R(k) einer bestimmten Art hat.

Dunham et al. beschreiben in dem US-Patent 4.791.565, "Apparatus for Controlling the Use of Computer Software", erteilt am 13. Dezember 1988, das Konstrukt der

"Zugangskontrolliste". In diesem Fall dienen die "Zugangsrechte" dazu, Lizenzbeschränkungen zu überwachen. Dunham verwendet einen Mikroprozessor auf EPROM-Basis als dedizierten Server. In dieser Anordnung werden Anforderungen zur Nutzung von Software, die von Terminals ausgehen und für einen Host-Rechner bestimmt sind, vor der Übertragung vermittelt. Jede Anforderung wird je nach den Kriterien, die in der Softwarelizenz der Benutzer genannt sind, entweder mit oder ohne Kommentar weitergeleitet oder zurückgewiesen.

Pailen et al. beschreiben in dem US-Patent 4.652.990, "Protected Software Access Control Apparatus and Method", erteilt am 24. März 1987, einen "Schloß-Schlüssel"-Ansatz zur Begrenzung des unerlaubten Kopierens. Bei Pailen wird mit Hilfe eines interaktiven Prozesses zur Erzeugung verschlüsselter Nachrichten zwischen einem anfordernden abgesetzten Terminal und einem Paar Vermittlungsprozessoren überprüft, ob Benutzer, Objekt und Rechte zueinander passen, bevor der Zugang gewährt wird.

Wolfe beschreibt im US-Patent 4.796.220, "Method of Controlling the Copying of Software", erteilt am 3. Januar 1989, einen weiteren Schloß-Schlüssel-Ansatz, bei dem Konfigurationsinformationen berechtigter Terminals im Rahmen einer Erlaubniscodeberechnung verwendet werden, die von einem Host an das anfordernde Terminal geschickt wird. Die Berechnung wird an die einzelnen Anforderungen angehängt und funktioniert bei späteren Zugriffen des Terminals auf den Host zusammen mit den Konfigurationsdaten als Schlüssel zur Neuberechnung des Codes.

Das IEEE-Papier von S. Vinter mit dem Titel "Extended Discretionary Access Controls" (Seite 39-49 der Proceedings of the 1988 IEEE Symposium on Security and Privacy, Oakland, California, 18.-21. April 1988, IEEE, New York, USA) beschreibt die Kontrolle der Zugangsberechtigung zu Ressourcen mit Hilfe von Zugangskontrollisten. Ein Client kann auf ein Objekt zugreifen, wenn seine Identität in einer Zugangskontrolliste erscheint, die mit einer Berechtigung zu dem angeforderten Zugang verbunden ist.

Unter einem Aspekt stellt die vorliegende Erfindung ein Verfahren zur Kontrolle des Zugangs zu Computerressourcen bereit, die sich in einem Host-Computer eines Computersystems befinden, das den Host und eine Anzahl M von Arbeitsstationen enthält, die zur Kommunikation mit dem Host verbunden sind, wobei das Verfahren folgende Schritt umfaßt:

(a) Aufrufen einer vorab berechneten Liste, wenn eine Arbeitsstation oder ein Benutzer den Zugang zu Ressourcen anfordern, wobei die Liste M Arbeitsstations- oder Benutzeridentitäten sowie eine verschlüsselte Darstellung der Zahl N der Arbeitsstationen oder Benutzer enthält, die zum Zugang zu den Ressourcen berechtigt sind, wobei N eine kleinere Zahl ist als M und die verschlüsselte Darstellung von N mit Hilfe eines Verschlüsselungsschlüssels als Funktion der Host-Identität und eines Versatzes gebildet wird;

(b) Feststellen der Tiefe N, bis zu der die Liste durchsucht werden kann, durch Entschlüsselung der verschlüsselten Darstellung des Parameters N mit Hilfe des Verschlüsselungsschlüssels; und

(c) Vergleichen der Identität der Arbeitsstation oder des Benutzers, von der bzw. dem die Diensteanforderung stammt, mit den Identitäten der M Arbeitsstationen oder Benutzer auf der Liste, jedoch nur bis zu einer Tiefe N, und Genehmigen des Zugangs, wenn eine Identitätsentsprechung gefunden wird, anderenfalls hingegen Zurückweisen der Zugangsanforderung.

Es wird angenommen, daß ein solches Verfahren gegen unbefugte Manipulationen gesichert ist.

In einem bevorzugten Ausführungsbeispiel stellt die vorliegende Erfindung ein gegen unbefugte Manipulationen gesichertes Verfahren zur Gewährung der Berechtigung zum Zugang zu Daten oder Anwendungssoftware zwischen einem Host und einer vorbestimmten Zahl N von M angeschlossenen Arbeitsstationen oder Benutzern, wobei N kleiner als M ist, der Host-Computer einen Kommunikationsserver für die Verwaltung der physischen Datenübertragung zwischen dem Host und M Arbeitsstationen oder Benutzern sowie ein Mittel zum Speichern der Zugangskontrollsoftware und zugehöriger Informationen umfaßt und das Verfahren beim Host folgende Schritte umfaßt:

Aufrufen einer vorab berechneten Liste, wenn eine Arbeitsstation oder ein Benutzer den Zugang zu Ressourcen anfordern, wobei

(a) Aufrufen von Zugangskontrollsoftware von dem Speichermittel sowie einer vorab berechneten Liste, wenn eine Arbeitsstation oder ein Benutzer einen Dienst anfordern, wobei die Liste M Stations- oder Benutzeridentitäten sowie eine verschlüsselte Darstellung von N für die Zahl der

Arbeitsstationen oder Benutzer enthält, die zum Zugang zu oder zu der Verbindung mit dem Host berechtigt sind, wobei die verschlüsselte Darstellung von N mit Hilfe eines Verschlüsselungsschlüssels als Funktion der Host-Identität und eines Versatzes gebildet wird;

(b) Feststellen der Tiefe N, bis zu der die Liste durchsucht werden kann, durch Entschlüsselung der Darstellung mit Hilfe des Schlüssels; und

(c) Vergleichen der Identität der Arbeitsstation oder des Benutzers, von der bzw. dem die Diensteanforderung stammt, mit den Identitäten der M Stationen oder Benutzer auf der Liste, jedoch nur bis zu einer Tiefe N, und Zurückgeben einer Berechtigung nur dann, wenn eine Identitätsentsprechung gefunden wird.

Es wird angenommen, daß eine solche Regelung ein gegen unbefugte Manipulationen gesichertes Verfahren zur Kontrolle der Zahl der Benutzer ist, die die Berechtigung zum Zugang zu lizenzierter Software in einem host-basierten System mit mehreren Terminals haben. Der Software für ein solches Verfahren kann in die Module integriert werden, die ein lizenziertes Softwareprodukt bilden.

Das obige Verfahren beruht auf der unerwarteten Verwendung einer verschlüsselten Form eines Parameters für die Tiefe der Berechtigungsliste. Wie nachstehend beschrieben wird, erfolgt die Berechtigung zum Zugang zu Daten zwischen einem Host und einer vorbestimmten Zahl N < M angeschlossener Arbeitsstationen oder Benutzer. Der Host umfaßt einen Kommunikationsserver für die Verwaltung der physischen

Datenübertragung zwischen dem Host und den M Arbeitsstationen oder Benutzern sowie ein Mittel zum Speichern der Zugangskontrollsoftware und zugehöriger Informationen.

Die erste Operation erfolgt beim Host und umfaßt das Aufrufen der Zugangskontrollsoftware von dem Speichermittel und das Aufrufen einer vorab berechneten Liste. Diese Aufrufe erfolgen beide, wenn eine Arbeitsstation oder ein Benutzer einen Dienst anfordern. Die Liste umfaßt M Stations- oder Benutzeridentitäten und eine verschlüsselte Darstellung des Parameters N. $N \leq M$ stellt die Zahl der Arbeitsstationen dar, die zum Zugang zu oder zu der Verbindung mit dem Host berechtigt sind.

Der Verschlüsselungsschlüssel ist eine Funktion der Hostidentität und eines Versatzes. Unter "Versatz" ist hier eine Konstante zu verstehen, die arithmetisch mit der Hostidentität kombiniert wird, um den Schlüssel zu verschleiern. Die Hostidentität könnte z.B. die Seriennummer des Hosts sein, die in dessen Speicher hart codiert ist, oder eine ganze Zahl, die additiv damit kombiniert wird.

Die zweite Operation umfaßt das Feststellen des Wertes des Tiefenparameters N durch Entschlüsseln der Darstellung mit Hilfe des Schlüssels. Der Wert N definiert die Tiefe, bis zu der die Liste durchsucht werden darf.

Die dritte Operation verlangt, daß die Identität des Diensteanforderers bis zu dieser Tiefe mit den Elementen der Liste verglichen wird und eine Berechtigung nur dann zurückgeben wird, wenn innerhalb dieser Tiefe eine Entsprechung festgestellt wird. Bedeutsam ist, daß jede

Änderung in der Suchtiefe N deren Neuverschlüsselung erfordert.

Vorteilhafterweise erfordert ein auf einem Host residentes lizenziertes Softwareprodukt, von dem ein Teil in darauf zugreifende Terminals heruntergeladen werden kann, mit dem Verfahren der vorliegenden Erfindung nur einen einzigen Installationsschritt außer der Regelung der Zahl der berechtigten Benutzer. Es erlaubt sogar die dynamische Berechtigung von Benutzern für ein einzelnes Gerät, denn der Verschlüsselungsschlüssel ist eine Funktion der Hostidentität. Zu beachten ist, daß die Nutzung der Hostidentität den Einsatz des Codes auf ein vorbestimmtes System beschränkt.

Die vorliegende Erfindung wird weiter exemplarisch unter Bezugnahme auf ein Ausführungsbeispiel beschrieben, das in den beiliegenden Zeichnungen dargestellt ist, wobei gilt:

Fig. 1 stellt ein System zum Herunterladen von der Host-CPU zur Arbeitsstation dar, und

Figs. 2-5 zeigen Beispiele 1-4 für Zugangskontrollisten.

In Fig. 1 sind eine CPU 1 und mehrere über die Bahnen 9, 11, 13, 15 mit ihr verbundene Terminals 17, 19, 21, 23 zu sehen. In der folgenden Beschreibung soll davon ausgegangen werden, daß der CPU-Knoten unter einem Betriebssystem läuft, das einen Kommunikationsserver ähnlich dem System verwendet, das entweder in "VM/System Product Programmer's Guide to the Server-Requester Programming Interface for VM/System Product" (S. 6-7), IBM-Publikation SC24-5291-1, Dezember 1986, oder in

"TSO Extensions Programmer's Guide to the Server-Requester Programming Interface for MVS/XA" (S. 1-3), IBM-Publikation SC28-1309-1, September 1987, beschrieben ist.

Andere Ressourcen von Recheneinrichtungen unterliegen den Betriebsgrundsätzen des IBM/370, wie sie im US-Patent 3.400.371, "Data Processing System", von Amdahl et al., ausgestellt am 3. September 1968, beschrieben sind.

Wie wiederum in Fig. 1 zu sehen ist, umfaßt die CPU 1 zusätzlich zu einer gewöhnlichen Ergänzung der Betriebssystemdienste vorzugsweise mindestens eine Anwendung, die in einer Kommunikationsbeziehung mit mindestens einem Terminal über eine Download-Schnittstelle zu einer Arbeitsstation ausgeführt werden kann, die über einen bezeichneten Pfad auf sie zugreift. Es ist zu berücksichtigen, daß lizenzierte Softwareprodukte in Form von reinem Objektcode (OCO, Object Code Only) ausgedrückt sind. Sie sind nach einer strukturierten Programmsyntax zusammengefaßt, die häufig mehrere Module mit einem einzelnen Eingang und einem einzelnen Ausgang umfaßt (siehe J. E. Nicholls, "The Structure and Design of Programming Languages", The Systems Programming Series, Copyright Addison-Wesley Publishing Co. 1975, Kapitel 12 zum modularen Programmieren, insbes. S. 486). Daher sind in dem bevorzugten Ausführungsbeispiel ein Zugangskontrollprogramm (ACP, Access Control Program) und eine Zugangskontrolliste (ACL, Access Control List) unter den Produktmodulen integriert. Sowohl die OCO-Produktform als auch die Verteilung von ACP und APL auf mehrere Module dürfte sie gegen eine Isolierung und zufällige Betrachtung relativ immun machen.

Zugangskontrolliste (ACL)

Die ACL umfaßt vorzugsweise eine Datei, die einen Vorspanndatensatz enthält, gefolgt von einem Datensatz pro berechtigtem Benutzer. Der Vorspanndatensatz gibt die Zahl der berechtigten Benutzer auf der Liste an. Wenn die Vorspanndatensätze z.B. einen verschlüsselten ganzzahligen Wert von drei enthalten, sind nur die ersten drei Benutzer auf der ACL berechtigt, die Download-Übertragungsoperation aufzurufen.

Um einem Benutzer die Berechtigung zu erteilen , muß auf die Datenmenge (das Modul) zugegriffen werden, die die ACL enthält und sich in der Host-CPU 1 befindet. An diesem Punkt kann im Einklang mit der im Vorspanndatensatz vorgeschriebenen Tiefe eine neue berechtigte ID eingegeben werden. Anzumerken ist hierzu, daß die Datenmenge zusätzlich geschützt werden kann, wie in der RACF (Ressource Control Facility, Ressourcenkontrolleinrichtung) von IBM beschrieben, die in der IBM-Publikation SC28-0733, "OS/VS2 MVS RACF Command Language Reference", dargestellt ist.

In den Figs. 2-5 sind Beispiele 1-4 für Zugangskontrollisten gemäß der vorliegenden Erfindung zu sehen. In Fig. 2 sind vier Namen mit einer Parametertiefe von N=3 aufgeführt. Somit sind nur die Terminal- oder Benutzeridentitäten GEORG, HANS und MARIE berechtigt, ROSA dagegen nicht. In Fig. 3 übersteigt die zulässige Tiefe die Länge der Liste, so daß eine weitere Identität hinzugefügt werden könnte. Fig. 4 zeigt eine Tiefe von 1, während Fig. 5 eine Liste mit einer anderen CPUID zeigt. Im letzten Fall würde der Tiefenparameter nicht entschlüsselt, da der Schlüssel eine

Funktion einer vorbestimmten CPUID zuzüglich eines Versatzes ist.

In der Praxis stützen sich die Berechtigungs- und Zugangsmechanismen unabhängig davon, ob die Host-CPU sich in einem lokalen Netz befindet oder eine VM mit angeschlossenen Terminals ist, vorwiegend auf einen Paßwortvergleich. Bei einer falschen Paßworteingabe oder bei wiederholt falscher Paßworteingabe wird einfach der Zugang verweigert. In anderen Systemen, wie z.B. in der bereits erwähnten RACF, können andere Kriterien für die Kontrolle des Zugangs zum System verwendet werden, wie z.B. der Standort oder ein Wert eines Systemtakts.

Zugangskontrollprogramm (ACP)

Es wird hier eine exemplarische Pseudocodesequenz mit starken PASCAL-Untertönen gezeigt, deren Ausführung das Verfahren der Erfindung beinhaltet. Bezeichnenderweise kann das ACP durch

```
ACP(userid: char, encrypt: bool) boolean
```

die Deklaration des ACP-Programms, entweder einmal pro angemeldeter Sitzung oder mehrmals aufgerufen werden (d.h. jedesmal, wenn eine Datenübertragung durchgeführt werden soll), wobei die Eingaben wie folgt definiert sind:

userid – eine Zeichenkette, die angibt, welche userid (Benutzerkennung) in der Zugangskontrolliste (ACL) gesucht werden soll

encrypt – boolesche Variable (TRUE/WAHR, wenn der ACL-Vorspann verschlüsselt ist, und FALSE/FALSCH, wenn der ACL-Vorspann entschlüsselt ist

ACL – Zugangskontrolliste (Access Control List)

Die Sequenz legt folgende Funktionen fest:

(a) Öffnen der Datei, die die ACL enthält.

```
Begin
     Reset (ACL)
```

(b) Lesen des Vorspanndatensatzes und Decodieren der Tiefenebene N

```
Read (ACL, header);
If (encrypt) then begin
     max_depth = decrypt (header, get_cpu_id)
End;
```

Implementiert wird dies durch Entschlüsseln des Vorspanns mit einem Schüssel, der nach irgendeinem bekannten Ver- und Entschlüsselungsalgorithmus aus der CPUID plus Versatz gebildet wird. Solche Algorithmen finden sich in Ehrsam et al., US-Patent 4.227.253, "Cryptographic Communication Security for Multiple

Domain Networks", erteilt am 7. Oktober 1980; Matyas et al., US-Patent 4.218.738, "Method for Authenticating the Identity of a User of an Information System", erteilt am 19. August 1980; sowie Meyer und Matyas, "Cryptography - New Dimension in Computer Data Security", Copyright John Wiley & Sons 1982.

```
Else begin
     max_depth = header
End;
```

Die Tiefenzahl im Vorspann ist klar.

(c) Durchsuchen der ACL nach einer Entsprechung zwischen der ID des Anforderers und der Liste innerhalb des verschlüsselten Tiefenbereichs N.

```
i = 0;
Not_found = TRUE;
While (i < max_depth) and (not_found) do begin
     Readin (ACL_userid)
     IF (ACL_userid = userid) then begin
          not_found = FALSE;
     End;
     i = i + 1;
End;
Return (not_found);
End;
```

(d) Wenn eine Entsprechung gefunden wird (d.h. wenn der zurückgegebene Wert (not_found) = FALSE ist), Aufrufen der Anwendung, für die die Berechtigung besteht, auf dem

Host. Anderenfalls (d.h. wenn der zurückgegebene Wert (not_found) = TRUE ist), Zurückgeben einer Nachricht an die anfordernde Arbeitsstation, daß keine Berechtigung besteht.

Wie der genannten Sequenz zu entnehmen sein dürfte, sind die zwei kritischen Strukturen der Bedingungsbefehl IF..THEN..ELSE zur Feststellung des Tiefenparameters, gefolgt von der Schleife WHILE..DO zum Durchsuchen der ACL nach einer Entsprechung.

A N S P R Ü C H E

1. Ein Verfahren zur Kontrolle des Zugangs zu Computerressourcen, die sich in einem Host-Computer eines Computersystems befinden, das den Host und eine Anzahl M von Arbeitsstationen enthält, die zur Kommunikation mit dem Host verbunden sind, wobei das Verfahren folgende Schritt umfaßt:

   (a) Aufrufen einer vorab berechneten Liste, wenn eine Arbeitsstation oder ein Benutzer den Zugang zu Ressourcen anfordern, wobei die Liste M Arbeitsstations- oder Benutzeridentitäten sowie eine verschlüsselte Darstellung der Zahl N der Arbeitsstationen oder Benutzer enthält, die zum Zugang zu den Ressourcen berechtigt sind, wobei N eine kleinere Zahl ist als M und die verschlüsselte Darstellung von N mit Hilfe eines Verschlüsselungsschlüssels als Funktion der Host-Identität und eines Versatzes gebildet wird;

   (b) Feststellen der Tiefe N, bis zu der die Liste durchsucht werden kann, durch Entschlüsselung der verschlüsselten Darstellung des Parameters N mit Hilfe des Verschlüsselungsschlüssels; und

   (c) Vergleichen der Identität der Arbeitsstation oder des Benutzers, von der bzw. dem die Diensteanforderung stammt, mit den Identitäten der M Arbeitsstationen oder Benutzer auf der Liste, jedoch nur bis zu einer Tiefe N, und Genehmigen des

Zugangs, wenn eine Identitätsentsprechung gefunden wird, anderenfalls hingegen Zurückweisen der Zugangsanforderung.

2. Ein Verfahren nach Anspruch 1 zur Kontrolle der Zugangsberechtigung für Anwendungsprogramme, wobei der Zugang zu einem Anwendungsprogramm das Aufrufen des Anwendungsprogramms umfaßt und bei der Zurückweisung einer Zugangsanforderung eine Zurückweisungsnachricht an die anfordernde Arbeitsstation oder den anfordernden Benutzer geschickt wird.

3. Ein Verfahren nach Anspruch 2, bei dem der Schritt des Aufrufens der Liste das Aufrufen von Zugangskontrollsoftware umfaßt, wobei die Liste und die Zugangskontrollsoftware in das Anwendungsprogramm integriert sind.

4. Ein Verfahren nach einem der obigen Ansprüche, bei dem die Anordnung des Hosts, der die Arbeitsstationen oder Benutzer kommunikativ verbindet, aus einer Menge ausgewählt wird, die aus einem lokalen Netz und einem Mehrprogramm-Mehrprozessor-System wie z.B. VM besteht.

5. Ein Verfahren nach einem der obigen Ansprüche, bei dem die Schritte des Verfahrens ferner das Modifizieren der Suchtiefe N allein durch deren Neuverschlüsselung umfassen.

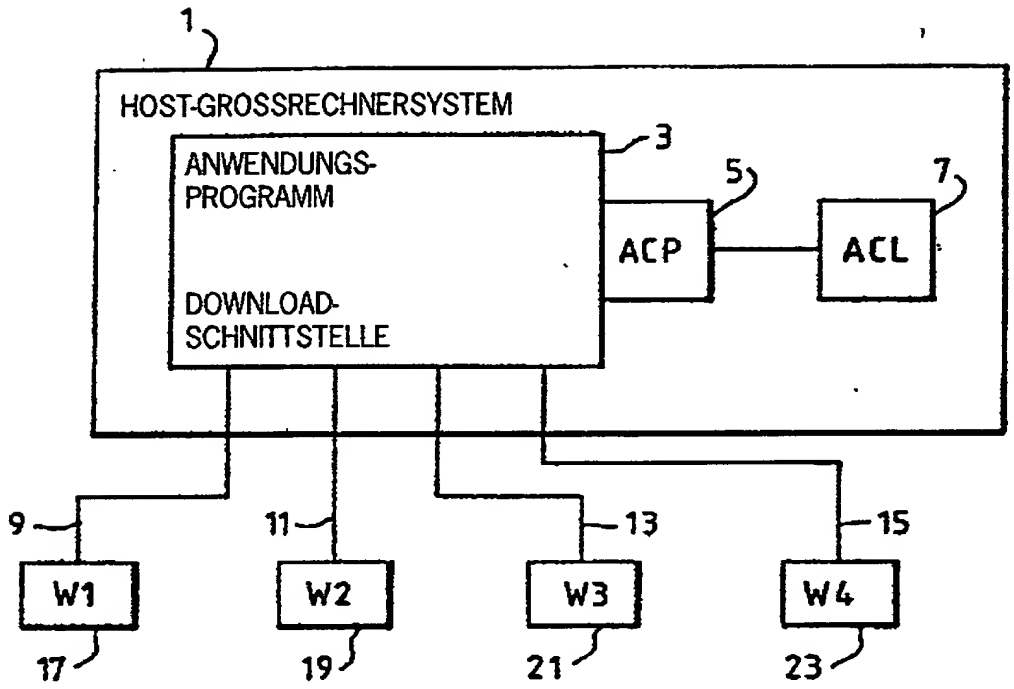
1
HOST-GROSSRECHNERSYSTEM
ANWENDUNGS-
PROGRAMM
DOWNLOAD-
SCHNITTSTELLE
3
5
7
ACP
ACL
9
11
13
15
W1
W2
W3
W4
17
19
21
23

Fig. 1

```
ACL-1
- VERSCHLÜSSELTER WERT 3 MIT AUSGANGSPUNKT=CPUID -
GEORG     /* BENUTZER 1, Z.B. W1 IN FIGUR 1 */
HANS      /* BENUTZER 2, Z.B. W2 IN FIGUR 1 */
MARIE     /* BENUTZER 3, Z.B. W3 IN FIGUR 1 */
ROSA      /* BENUTZER 4, Z.B. W4 IN FIGUR 1 */
```

ZUGANGSKONTROLLISTE, BEISPIEL 1

Fig. 2

```
ACL-2
- VERSCHLÜSSELTER WERT 5 MIT AUSGANGSPUNKT=CPUID -
GEORG     /* BENUTZER 1, Z.B. W1 IN FIGUR 1 */
HANS      /* BENUTZER 2, Z.B. W2 IN FIGUR 1 */
MARIE     /* BENUTZER 3, Z.B. W3 IN FIGUR 1 */
ROSA      /* BENUTZER 4, Z.B. W4 IN FIGUR 1 */
```

ZUGANGSKONTROLLISTE, BEISPIEL 2

Fig. 3

```
ACL-3
1
GEORG     /* BENUTZER 1, Z.B. W1 IN FIGUR 1 */
HANS      /* BENUTZER 2, Z.B. W2 IN FIGUR 1 */
MARIE     /* BENUTZER 3, Z.B. W3 IN FIGUR 1 */
ROSA      /* BENUTZER 4, Z.B. W4 IN FIGUR 1 */
```

ZUGANGSKONTROLLISTE, BEISPIEL 3

Fig. 4

```
ACL-4
- VERSCHLÜSSELTER WERT 3 MIT AUSGANGSPUNKT=ANDERE CPUID ALS HOST-CPUID -
GEORG     /* BENUTZER 1, Z.B. W1 IN FIGUR 1 */
HANS      /* BENUTZER 2, Z.B. W2 IN FIGUR 1 */
MARIE     /* BENUTZER 3, Z.B. W3 IN FIGUR 1 */
ROSA      /* BENUTZER 4, Z.B. W4 IN FIGUR 1 */
```

ZUGANGSKONTROLLISTE, BEISPIEL 4

Fig. 5